Journal of Telecommunication, Electronic and Computer Engineering
ISSN: 2180 – 1843 e-ISSN: 2289-8131 Vol. 17 No. 3 (2025) 1-6
jtec.utem.edu.my
DOI: https://doi.org/10.54554/jtec.2025.17.03.001

A Hybrid Genetic Algorithm and Tabu Search for Construction Site Layout Planning Problem

Trieu Xuan Hoa[1], Tranh Thanh Thuong[2]*
[1]*Thai Nguyen University of Agriculture and Forestry, Vietnam*
[2]*International School, Thai Nguyen University, Vietnam*

Article Info

Article history:
Received Dec 18th, 2024
Revised June 23rd, 2025
Accepted Sep 24th, 2025
Published Sep 30th, 2025

Index Terms:
CSLP problem
GA
Tabu Search
GA-Tabu

Abstract

The construction site layout is widely acknowledged as a complex challenge in the field of construction management. However, it is crucial for nearly all construction projects, as the layout of site facilities can significantly impact on the project costs. This issue can be formulated as a Quadratic Assignment Problem, which is widely recognized to be NP-hard problem. While numerous methods have been proposed, most existing approaches either lack sufficient global search capabilities or struggle to escape local optima, leading to suboptimal solutions. To address these limitations, this study proposes a hybrid GA-Tabu algorithm that integrates the exploration strength of Genetic Algorithms with the exploitation efficiency of Tabu Search. This integration aims to achieve a better balance between global and local search processes, thereby enhancing the overall solution quality for the CSLP problem. The proposed algorithm has been subsequently tested with three case studies previously utilized in related research, effectively demonstrating its performance and applicability.

*Corresponding Author: tranthanhthuong@tnu.edu.vn

I. INTRODUCTION

The problem of site layout planning is considered as one of the most complex and challenging issues in the field of construction management. Construction site layout planning (CSLP) involves the optimal allocation of a given set of n predefined facilities to a set of m predetermined vacant locations, where $m \geq n$ [1]. This optimization process aims to enhance spatial efficiency, minimize operational conflicts, and improve overall site functionality.

Lam et al. emphasized that the optimizing material handling costs by 20-60% depends on the adoption of a suitable facility layout [2]. Consequently, meticulous construction site layout planning is a critical factor in the success of construction projects [3]. A well-designed construction site layout not only enhances operational efficiency and effectiveness but must also navigate various constraints, including spatial limitations, neighboring structures, site access, and specific positioning alignment requirements of the forthcoming edifice [2]. Due to the complexity of layout planning, construction managers frequently rely on past experiences, heuristic guidelines, or a first-come-first-serve approach: methods that frequently lead to inefficiencies [1], [4].

The primary objective of CSLP is to strategically position temporary facilities such as job offices, labor residences, warehouses and batch plants [1] in a way that fulfills design requirements while optimizing layout preferences, such as minimizing the total interaction cost between these facilities [5]. CSLP can be formulated as a Quadratic Assignment Problem (QAP) when the cost of interactions between different facilities is considered to be directly proportional to both the distance traveled and the volume of flow between facilities, as described by Tate and Smith [6], [7]. QAPs are classified as NP-hard due to their computational intensity, making it impractical to obtain exact solutions for realistic layout sizes [5]. The combinatorial nature of the problem leads to factorial growth in potential configurations, where n facilities yield $n!$ possible arrangements, which increases at a rate exceeding e^n. This exponential growth leads to an immense number of potential solutions, even for relatively small values of n, posing significant computational challenges [8]. An example is illustrated below.

Table 1
Facilities' number and possible alternatives

# Facilities (n)	# Feasible Configurations (n!)
9	362,880
10	3,628,800
11	39,916,800
12	479,001,600
15	1,307,674,368,000

As shown above, the number of feasible configurations increases significantly. Specifically, with just 15 facilities, the number of feasible configurations reaches the order of 10^13. Notably, a project involving 15 facilities (n = 15) is still considered relatively small in scale [9].

Solving CSLP remains challenging due to its combinatorial complexity, spatial and operational constraints, and the need for real-time responsiveness in dynamic site conditions. Balancing solution quality with computational feasibility is a key concern addressed in this study.

Several methods have been proposed to solve CSLP problem; however, many struggle with premature convergence or inadequate exploration of the solution space. Techniques such as Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) show promise but are often sensitive to initial parameters. To overcome these limitations, this study proposes a hybrid GA-Tabu algorithm that combines the global search capabilities of Genetic Algorithms with the local refinement and cycle avoidance features of Tabu Search.

This research seek to answer the following questions: (1) Can the proposed GA-Tabu hybrid algorithm improve layout optimization compared to existing state-of-the-art methods? (2) How does GA-Tabu perform across varying CSLP scenarios with different spatial and constraint configurations? (3) What are the practical implications of applying GA-Tabu to real-world construction planning?

To address these questions, the proposed algorithm was tested on three representative case studies. Results demonstrated its effectiveness, with performance comparable to or better than benchmark methods such as lopt-aiNet. These findings highlight GA-Tabu's potential as a practical and efficient tool for optimizing construction site layouts.

II. LITERATURE REVIEW

Several approaches have been developed to address the site layout planning problem, among which approximate algorithms are the most widely used methods.

Ant Colony Optimization (ACO) algorithms have been utilized to address site layout issues in construction projects [10], [11]. This approach has been enhanced by incorporating a Genetic Algorithm that integrates the Max-Min Ant System into the initialization phase of the GA [12]. Calis and Yuksel also employed ACO for the layout planning of temporary construction facilities [13]. In a subsequent development, they introduced a meta-heuristic algorithm named ACO-PA, which combines ACO with Parametric Analysis [14].

Particle Swarm Optimization (PSO) is another popular method for addressing the CSLP problem. Zhang et al. introduced a PSO-based approach to tackle this issue [15]. Lien and Cheng presented the particle-bee algorithm (PBA), which integrates the advantageous behaviors of honey bee and bird swarms to enhance search capabilities.

Adrian introduced a method that employs three metaheuristics: Genetic Algorithm, Ant Colony and Particle Swarm Optimizations, to address the CLS problem, aiming to minimize total construction and interaction costs arising from layout constraints [16]. Initially, the author intended to develop a hybrid local search-based aiNet algorithm, but later found that Vu had already developed a similar approach named lopt-aiNet [17], which demonstrated efficiency compared to previous methods.

Recent advancements in metaheuristics have led to the development of hybrid algorithms that combine classical search strategies with learning components. For instance, Wang [18] used deep-learning-enhanced genetic algorithms for complex scheduling tasks, while Xu and Li [19] applied an improved NSGA-II to address multi-objective construction site layout planning, effectively balancing spatial constraints and cost efficiency using Pareto-based optimization. Moreover, Monteiro et al. proposed a hybrid framework for multi-objective CSLP, which improved spatial efficiency and productivity [20]. These developments highlight the growing relevance of hybrid models like GA-Tabu in solving complex layout optimization problems.

Despite these advancements, most prior studies rely on standalone metaheuristics, which either lack diversity in the population or face challenges with local refinement. To our knowledge, few studies have combined GA with Tabu Search for CSLP. Our proposed hybrid method uniquely combines global and local search capabilities, aiming to reduce the risk of premature convergence while maintaining solution diversity.

III. METHODOLOGY

A. Problem Formulation

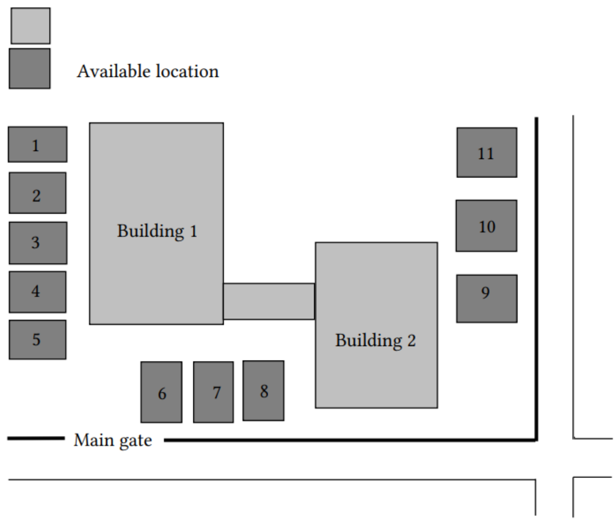

Figure 1. Location representation of the construction site

This study adopts a formulation for the CSLP problem that aims to minimize transportation costs between facilities. The formulation follows the approach described in [17] as below:

Assuming the number of predefined facilities and available locations is equal to n, if the number of locations exceeds the number of facilities, additional "dummy" facilities with zero distance and frequency can be introduced to maintain numerical consistency. Facilities are interconnected to reflect realistic movement patterns—for example, frequent travel typically occurs between the site office and the concrete batch workshop, while trips from the site office to the storeroom are less common. Therefore, facility locations are strategically selected to minimize transportation costs, which constitutes the primary objective of this problem. The total transportation distance is formulated in Equations (1) and (2).

$$Min\ F = \sum_{i=1}^{n} \sum_{x=1}^{n} \sum_{j=1}^{n} \delta_{xi}\, f_{xi} d_{ij} \quad (1)$$

Subject to

$$\sum_{i=1}^{n} \delta_{xi} = 1,\ \{i=1, 2, ., n\} \quad (2)$$

Where n is the number of facilities; δxi is the permutation matrix variable. Coefficient fxi is the frequency of trips made by construction personnel between facilities x and i. From the definition, it can be observed that fij equals to fji. Frequency is defined as the number of trips within a given time period, specifically defined in this study as the number of trips per day. Coefficient dij is the distance between locations i and j. Therefore, the objective function, F, represents the total travel distance covered by construction personnel [14]. Figure 1 presents a representative example with n=11.

One challenge in the dataset is the assumption of fixed and symmetric flow and distance values, which may not fully capture dynamic or asymmetric real-world conditions. Additionally, variations in facility size must be carefully encoding as constraints to avoid infeasible assignments.

B. Genetic Algorithm

Genetic algorithms (GAs) are modeling techniques based on biological behavior. This approach mimics natural selection and genetic inheritance [21]. In nature, individuals that are healthy and well-adapted to their environment are more likely to reproduce and pass on their traits to the next generation. Each individual has a unique genetic structure that determines its characteristics. During reproduction, offspring may inherit traits from both parents, and occasional mutations may introduce new genes not present in either parent.

GA operates on a population of individuals, where each population represents a generation. From the initially generated generation, GA applies natural selection and genetic processes to evolve the population. The main processes involved in designing and implementing GA include selection, crossover and mutation operations [22] as described below:

Selection

Good individuals are selected for the next generation based on their adaptability to the environment. The adaptability of each individual is measured using a fitness function.

Crossover

This process simulates genetic recombination. A father and mother exchange segments of their genetic code to produce two new offspring. Crossover is applied with a probability of p_c. This probability determines the number of individuals selected for this operation, calculated as $p_c.n$, where n is the population size.

Mutation

This introduces variation by randomly altering one or more genes in an individual's chromosome. After the crossover step, mutation is applied with a low probability of p_m to prevent premature convergence and maintain diversity.

After these steps, a new generation is formed. These operations are repeated until the termination condition is satisfied. The general structure of the GA algorithm is as follows:

Procedure Genetic_Algorithm;
Begin
    t ← 0;
    Initialize the first-generation P(t);
    Evaluate P(t) (using fitness function)
Repeat
    t ← t + 1;
    Create new generation P(t) from P(t−1) by:
        Selection;
        Crossover;
        Mutation;
    Evaluate P(t);
Until a stopping criteria has been met;
End.

In each cycle, GA begins with a randomly generated population and produces the next generation using three basic operators :

Selection: Poor-performing individuals from the population are removed. The population is sorted in descending order based on fitness score, and the good n individuals are retained.

Crossover: Selected individuals are represented as binary strings. A crossover point is randomly selected, and two new offspring are formed by exchanging segments of the parents' genetic material at the crossover.

- Mutation: A newly formed individual may undergo mutation at a randomly selected bit, changing a 0 to 1 or vice versa, with a small probability.

The evolutionary approach is repeated until a satisfactory solution is found. It enables efficient exploration of the solution space while maintaining diversity within the population.

C. Tabu search

Tabu search, created and formalized by Fred W. Glover [23], [24], [25] is a metaheuristic optimization approach that incorporates local search techniques for solving mathematical problem. Starting from an initial solution, TABU Search repeatedly explores the solution space to gradually improve the best available solution (referred to as a record).

At each iteration, the algorithm traverses the neighborhood (or the entire neighborhood) of the current solution to choose the best neighboring solution, which replaces the current solution in the next iteration. Each neighboring solution is generated by applying a modification known as a move. The set of all such solutions is called the neighborhood of the current solution.

The general Tabu Search procedure is as follows:

Procedure TABUSEARCH;
Begin
    Create the initial solution;
    Tabulist = ∅;
    While (A stopping criteria hasn't been met) do
        begin
            M = ∅;
            For each s' ∈ N_s
                begin
                    p = move(s, s'); // Create set neighborhood solutions of M
                    if (p ∉ Tabulist and p meets the requirement)
                        M = M ∪ p;
                end;
            Select the best p in M;
            Create s' from p of s;
            s = s';
            if (s' is better than sbest) sbest = s';
            Update Tabulist;
            Diversify the solution;
            Strengthen the solution;
        end;
    Return sbest;
End;

D. GA-Tabu - A Hybrid Algorihm of GA and Tabu Search

To improve the efficiency of GA, Tabu Search is incorporated to efficiently identify local optima. This hybrid approach, referred to as GA-Tabu (GATA), leverages the exploration ability of GA and the exploitation strength of Tabu Search, ultimately enhancing the convergence speed and solution quality for the CSLP problem.

In this study, a chromosome represents a complete layout of facilities on a construction site. Each gene in the chromosome corresponds to a specific facility, and its value indicates the assigned location. For a problem with *n* facilities and *n* available locations, the chromosome is encoded as a permutation of *n* integers. This encoding ensures that each facility is assigned a unique position, maintaining feasibility.

Parent chromosomes are selected using roulette wheel selection based on their fitness values. This probabilistic method favors individuals with better fitness while maintaining diversity in the population. The crossover operation utilizes the Partially Mapped Crossover (PMX) technique, suitable for permutation-based representations. PMX ensures that offspring are also valid permutations (i.e., no duplicate or missing locations). The crossover probability (p_n) is set to 0.8, meaning 80% of the selected pairs undergo crossover to produce new offspring. The mutation operator employs swap mutation, in which two random positions in the chromosome are swapped to introduce variability and prevent premature convergence. The mutation probability (p_m) is set to 0.1.

The objective is to minimize the total cost of the construction site layout, which consists of the material handling cost and travel cost between facilities. The objective function is defined as follows:

$$\text{Minimize } Z = \sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}\, d_{ij} \tag{3}$$

Where:

f_{ij} = the flow of materials or interactions between facilities *i* and *j*

d_{ij} = the distance between the locations assigned to facilities *i* and *j*

To improve solution quality and avoid premature convergence, Tabu Search is applied as a local search enhancement within the GA process. After each generation is produced through selection, crossover, and mutation, a small number of top-performing individuals (elites) are selected for refinement. Each elite solution undergoes Tabu Search for a fixed number of iterations, exploring neighboring solutions by swapping facility positions. Moves recently performed are stored in a tabu list to prevent cycling, unless they meet aspiration criteria. If a better solution is found during this search, it replaces the original in the population. This hybrid approach allows GA to perform global exploration while Tabu Search intensifies the search locally, achieving better balance and faster convergence. The proposed GATA solution is described as below:

Procedure GA_Tabu
Begin
  t ← 0;
  Initialize the first generation P(t) with n individuals;
  Evaluate P(t) (using fitness function);
Repeat
  t ← t + 1;
  Create new generation P(t) from P(t−1) by:
    Selection: Select n_1 best individuals in P(t−1);
    Crossover(n_1): The father and mother exchange genes to create two off-spring;
    Mutation(n_1): The mutation rate is inversely proportional to the adaptability, selecting a number of high adaptability individuals to put into a memory set;
    TabuSearch(n_1);
    Insert randomly individuals into P(t);
  Evaluate P(t);
Until a stopping criteria has been met;
End;

Where the Tabu Search is implemented as follows:

Procedure TabuSearch(M);
Begin
  Repeat
    1) Create set neighborhood solutions of M;
    2) Determine objective function for each neighborhood;
    3) Choose best neighborhood;
    4) Update Tabu list;
  Until A stopping criteria has been met;
end;

In the experimental setup, the hybrid GA-Tabu algorithm was configured with a population size of 100 individuals evolved over 20 generations, resulting in a total of 2,000 candidate solutions evaluated. The crossover and mutation probabilities were set to 0.8 and 0.1, respectively, following common practices in genetic algorithm design. For the Tabu Search component, a neighborhood size of 10 was used per iteration, with a tabu list size of 7 to prevent cycling. The algorithm terminated after reaching 2,000 iterations or when no further improvement was observed.

IV. RESULTS AND DISCUSSION

A computer with a Core i5 2.7 Ghz CPU and 8GB RAM was used for testing. Three case studies from the literature were selected to evaluate the hybrid algorithm. These case studies were chosen because they address the same CSLP problem, making them suitable benchmarks for comparison. The dataset used in each case study comprised 11 facilities and 11 locations. The interaction frequency matrix and distance matrix are both symmetric 11×11 matrices. Case Studies 2 and 3 reuse the base dataset of Case Study 1, with additional spatial or assignment constraints to simulate different practical limitations.

Case 1 assumes that any predefined location is available to accommodate any facility [10]. Tables 2 and 3 provide details on the facilities assigned within the site boundaries. The site layout does not include alternative routes between locations, and distances are measured in meters. Additionally, Table 3 outlines the daily trip frequencies between facilities. As shown in the table, the trip frequency between any two facilities is identical in both directions, resulting in a symmetric matrix. For instance, Table 3 shows that the daily trips between the Labor Residence and the Carpentry Workshop are 4. Case Studies 2 and 3 were created based on Case Study 1.

Case 2 assumes that the Side gate is designated to location 1 and the main gate is designated to location 10 [26]. This setup reflects a practical approach commonly used in construction projects, where gate locations are determined before the start of construction. Since gates play a crucial role in site access and transportation, their placement must be fixed at predefined locations.

Case 3 assumes that Site Office, Labor Residence Concrete Batch Shop cannot be assigned to the comparatively smaller locations 7 and 8 [27]. This scenario highlights a constraint where larger facilities cannot be placed in every available location due to size limitations. To address this, an unequal area constraint is applied to prevent the allocation of larger facilities to undersized locations.

The three case studies represent diverse real-world scenarios: unconstrained allocation (Case 1), fixed gate constraints (Case 2), and size-restricted placement (Case 3). The GA-Tabu algorithm maintained high performance across all scenarios, demonstrating its adaptability. It consistently found optimal or near-optimal layouts while respecting each case’s unique constraints, showcasing robustness across varying construction project conditions.

The experiment results are shown in Table 4 and are compared with those of some previous approaches in Figure . Table 4 presents the total transportation costs calculated using Equation (1), which quantified the objective function value for each candidate site layout. These values are expressed in arbitrary cost units based on actual distances and frequencies of travel between facilities. Each case was executed 10 times to account for the stochastic nature of the algorithm, and both the best and average results were reported. Additionally, the total CPU time in milliseconds for each run was included in the table

Figure 2 shows a comparison among GA-Tabu and the results of GA [26], PSO [15], ACO [10], ACO-PA [13], opt-aiNEt and lopt-aiNet [17]. It could be observed that the value of GA-Tabu is identical to that of lopt-aiNet and surpasses the other methods mentioned. In Case 2 and Case 3, the outcome achieved by GA-Tabu is superior to the others by 3.1% and 1.6%, respectively, as compared to the results in [17]. This highlights the efficiency of the GA-Tabu algorithm.

Although the GA-Tabu algorithm has demonstrated strong performance, several promising directions remain for future work. These include adaptive parameter tuning (e.g., dynamic crossover and mutation rates), hybridization with reinforcement learning for enhanced decision-making, and integration of real-time Building Information Modeling (BIM) data to enable dynamic layout adjustments during project execution. Furthermore, applying GA-Tabu to larger-scale construction site layouts, incorporating time-dependent facility interactions, and extending the algorithm to support 3D spatial planning would further enhance its applicability. To address scalability challenges in real-world scenarios, future research may also explore parallel or cloud-based implementations.

Table 2
The frequency of trips between facilities one day

	SO	FS	LR	S1	S2	CW	RW	SG	UR	BW	W
SO	0	5	2	2	1	1	4	1	2	9	1
FS	5	0	2	5	1	2	7	8	2	3	8
LR	2	2	0	7	4	4	9	4	5	6	5
S1	2	5	7	0	8	7	8	1	8	5	1
S2	1	1	4	8	0	3	4	1	3	3	6
CW	1	2	4	7	3	0	5	8	4	7	5
RW	4	7	9	8	4	5	0	7	6	3	2
SG	1	8	4	1	1	8	7	0	9	4	8
UR	2	2	5	8	3	4	6	9	0	5	3
BW	9	3	6	5	3	7	3	4	5	0	5
W	1	8	5	1	6	5	2	8	3	5	0

Table 3
The distances between available locations

	1	2	3	4	5	6	7	8	9	10	11
1	0	15	25	33	40	42	47	5	35	30	20
2	15	0	10	18	25	27	32	42	50	45	35
3	25	10	0	8	15	17	22	32	52	55	45
4	33	18	8	0	7	9	14	24	44	49	53
5	40	25	15	7	0	2	7	17	37	42	52
6	42	27	17	9	2	0	5	15	35	40	50
7	47	32	22	14	7	5	0	10	30	35	40
8	5	42	32	24	17	15	10	0	20	25	35
9	35	50	52	44	37	35	30	20	0	5	15
10	30	45	55	49	42	40	35	25	5	0	10
11	20	35	45	53	52	50	40	35	15	10	0

Table 4
Experimental results of 3 case studies

	Case 1		Case 2		Case 3	
Test	Result	Time	Result	Time	Result	Time
1	12232	3003	12232	3843	12404	4516
2	12150	3371	12150	3910	12404	3043
3	12150	3166	12202	3216	12404	3139
4	12150	2939	12345	2956	12404	2899
5	12150	2831	12150	2811	12404	2999
6	12150	3003	12280	2780	12404	3132
7	12150	3281	12232	2769	12404	2952
8	12150	3349	12150	2918	12404	3081
9	12232	3003	12246	3122	12404	3155
10	12150	4195	12202	2852	12404	3407
Best	12150	2831	12150	2769	12404	2899
Average	12166.4	3214.1	12218.9	3117.7	12404	3232.2

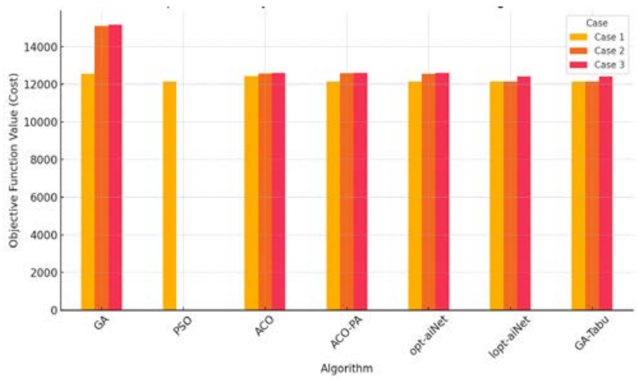

Figure 2. Comparison of GA-Tabu to exisisting methods

V. CONCLUSION

The study presents a novel algorithm that combines Genetic Algorithms and Tabu Search to address the NP-hard Construction Site Layout Planning (CSLP) problem. Across three case studies, the algorithm exhibited commendable performance. Notably, the effectiveness of the GA-Tabu algorithm was highlighted by its comparable performance to lopt-aiNet's performance and its superior outcomes compared to other existing models. In particularly, for Case 2 and Case 3, GA-Tabu achieved significant improvements, with reductions of 3.1% and 1.6%, respectively, compared to previous works. These results confirm the algorithm's efficiency and suggest its potential for solving larger and more complex CSLP problems. While GA-Tabu has shown strong performance, future work will focus on adaptive parameter tuning, integration with reinforcement learning, and incorporation of real-time BIM data to support dynamic layout adjustments. Additionally, expanding the algorithm to handle larger-scale and 3D site layouts, as well as exploring parallel or cloud-based implementations, will be considered to improve scalability and practical applicability.

CONFLICT OF INTEREST

Authors declare that there is no conflict of interests regarding the publication of the paper.

AUTHOR CONTRIBUTION

The author contribution to the paper as follows: study conception and design: Trieu Xuan Hoa; data collection: Trieu Xuan Hoa; analysis and interpretation of findings: Trieu Xuan Hoa, Tran Thanh Thuong; draft manuscript preparation: Trieu Xuan Hoa, Tran Thanh Thuong. All authors had reviewed the findings and approved the final manuscript.

REFERENCES

[1] I. N. Papadaki, and A. P. Chassiakos, “Multi-objective construction site layout planning using genetic algorithms,” Procedia Engineering, vol. 164, pp. 20-27, 2016.

[2] K.C. Lam, C. M. Tang, and W. C. Lee, “Application of the entropy technique and genetic algorithms to construction site layout planning of medium-size projects,” Construction Management and Economics, vol. 23, no. 2, pp. 127-145, 2005.

[3] X. Ning, K. Lam, and M. C. Lam, “Dynamic construction site layout planning using max-min ant system,” Automation in Construction, vol. 19, no. 1, pp. 55-65, 2010.

[4] H. Said, and K. El-Rayes, “Performance of global optimization models for dynamic site layout planning of construction,” Automation in Construction, vol. 36, pp. 71-78, 2013.

[5] I. Yeh, “Architectural layout optimization using annealed neural network,” Automation in Construction, vol. 15, no. 4, pp. 531-539, 2006.

[6] D. M. Tate, and A. E. Smith, “Unequal-area facility layout by genetic search,” IIE Transactions, vol. 27, no. 4, pp. 465-472, 1995.

[7] A. Azarbonyad, and R. Babazadeh, “A genetic algorithm for solving quadratic assignment problem (QAP),” in Proc. of 5th International Conference of Iranian Operations Research Society (ICIORS), 2012, https://doi.org/10.48550/arXiv.1405.5050

[8] A. Kaveh, M. Khanzadi, M. Alipour, and M. R. Moghaddam, “Construction site layout planning problem using two new metaheuristic algorithms,” Iranian Journal of Science and Technology, Transactions of Civil Engineering, vol. 40, pp. 263-275, 2016.

[9] I. Yeh, “Construction-site layout using annealed neural,” Journal of Computing in Civil Engineering, vol. 9, no. 3, pp. 201-208, 1995.

[10] E. Gharaie, A. Afshar, and M. R. Jalali, “Site layout optimization with ACO algorithm,” in Proc. of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, 2006, pp. 90-94.

[11] K. Lam, X. Ning, and T. Ng, “The application of the ant colony optimization algorithm to the construction site layout planning problem,” Construction Management and Economics, vol. 25, no. 4, pp. 359-374, 2007.

[12] K. C. Lam, X. Ning, and M. C. Lam, “Conjoining MMAS to GA to solve construction site layout planning problem,” Journal of Construction Engineering and Management, vol. 135, no. 10, pp. 1049-1057, 2009.

[13] G. Calis, and O. Yuksel, “A comparative study for layout planning of temporary construction facilities: optimization by using ant colony algorithm,” in Proc. of 13th International Conference on Computing in Civil and Building Engineering, 2010.

[14] G. Calis and O. Yuksel, “An improved ant colony optimization algorithm for construction site layout problems,” Journal of Building Construction and Planning Research, vol. 3, no. 4, 2015.

[15] H. Zhang, and J. Y. Wang, “Particle swarm optimization for construction site unequal-area layout,” Journal of Construction Engineering and Management, vol. 134, no. 9, pp. 739-748, 2008.

[16] A. M. Adrian, A. Utamima, and K. Wang, “A comparative study of GA, PSO and ACO for solving construction site layout optimization,” KSCE Journal of Civil Engineering, vol. 19, no. 3, p. 520–527, 2015.

[17] D. Vu, T. N. Van, and H. H. Xuan, “An improved artificial immune network for solving construction site layout optimization,” in Proc. of 2016 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2016.

[18] H. W. a. H. Z. C. Ma, “A hybrid particle swarm optimization for construction site layout planning with safety considerations,” Automation in Construction, vol. 124, 2021.

[19] X. Xu and H. Li, “Multi-objective construction site layout planning using improved NSGA-II,” Engineering Applications of Artificial Intelligence, vol. 110, 2022.

[20] M. L. A. E. Borges, and A. D. Granja, and A. Monteiro, “A hybrid framework for multi-objective construction site layout optimization,” Buildings, vol. 14, no. 12, 2024.

[21] P. C. Chu, and J. E. Beasley, “A genetic algorithm for the generalised assignment problem,” Computers & Operations Research, vol. 24, no. 1, pp. 17-23, 1997.

[22] M. J. Mawdesley, S. H. Al-jibouri, and H. Yang, “Genetic algorithms for construction site layout in project planning,” Journal of Construction Engineering and Management, vol. 128, no. 5, pp. 418-426, 2002.

[23] F. Glover, “Future paths for integer programming and links to artificial intelligence,” Computers & Operations Research, vol. 13, no. 5, pp. 533-549, 1986.

[24] F. Glover, “Tabu search—part I,” ORSA Journal on Computing, vol. 1, no. 3, pp. 190-206, 1989.

[25] F. Glover, “Tabu search—part II,” ORSA Journal on Computing, vol. 2, no. 1, pp. 4-32, 1990.

[26] L. P. E. Love, “Comparing genetic algorithms and non-linear optimisation for labor and equipment assignment,” Computing in Civil Engineering, vol. 12, pp. 227-231, 1998.

[27] H. Li, and P. E. D. Love, “Genetic search for solving construction site-level unequal-area facility layout problems,” Automation in Construction, vol. 9, no. 2, pp. 217-226, 2000.